An Overload Control Algorithm based on Priority Scheduling for SIP Proxy Server

Inwhee Joe, Janghyun Lee
Division of Computer Science and Engineering, Hanyang University, Seoul, 133-791, South Korea

Abstract—For real-time multimedia services, SIP becomes one of the most popular. Most SIP entities use an unreliable Transport Protocol (UDP). For unreliable transport protocol, SIP introduces a retransmission mechanism to ensure the reliability of its real-time message delivery. However, retransmission mechanism makes an overload worse when SIP Proxy server is overloaded. Existing SIP overload control algorithms provide limited congestion control with the using 503 service unavailable message and thresholds of proxy server. In this paper, we propose a new overload control algorithm based on priority scheduling to improve a call session setup delay and the throughput. Also, we compare our algorithm with the previous overload control algorithm in terms of the call setup delay and throughput. The performance results show significant improvement over existing algorithms.

Keywords: component, SIP, UDP, priority scheduling

1. Introduction

The Session Initiation Protocol (SIP) is an application layer signaling protocol that can establish, modify, and terminate multimedia sessions. It can be used to create two-party, multiparty, or multicast sessions that include Internet telephone calls, multimedia distribution, and multimedia conferences [1].

The SIP protocol is an Application Layer protocol designed to be independent of the underlying Transport Layer. Because SIP becomes one of most popular protocols in real-time multimedia services, Most SIP entities use an unreliable Transport Protocol (UDP) for signaling [5]. RFC3261 SIP introduces a retransmission mechanism to ensure the reliability of its real-time message delivery for UDP [3].

However, retransmissions make an overload worse when SIP Proxy server is overloaded. Because of the excessive growth of internet based multimedia services, SIP overload control has been studied as an important issue.

Currently SIP provides limited support for overload handling through its 503 response code, which tells an upstream element that it is overloaded. However, numerous problems have been identified with this mechanism.

In this paper, we propose a new overload control algorithm based on priority scheduling to improve a call session setup delay and the throughput.

The rest of the paper is organized as follows. In Section II, We present related works of sip overload control, and in Section III, we propose a sip overload control method. The performance evaluation results for the proposed method are given in Section IV. We conclude the paper in Section V.

2. Related works

SIP Components are basically composed of SIP UAs and SIP Server. UAs, which include UAC (User Agent Client) and UAS (User Agent Server), participate as clients which make SIP call process. SIP servers include proxy server for session routing and registration server for UA registration.

Figure 1 shows the SIP call message flow for session establishment. UAC first sends INVITE request message towards the SIP Proxy Server. Proxy Server responds to an INVITE request message as 100Trying message and decides next hop to forwarding.

When UAS receives an INVITE request message, it returns 180Ringing and 200OK. Finally UAC acknowledges with an ACK message. Then, a session is established between UAC and UAS.

![Figure 1. Session establishment flow](image)

In the case of Unreliable transmission Protocol (UDP), SIP RFC3261 introduces a retransmission mechanism to ensure the reliability of its real-time message delivery.

When UAC does not receive 100Trying, it retransmits the INVITE request message by timer T1. Timer T1, which is an estimate of the round trip time, increases exponentially...
until it reaches 32s. If Timer T1 exceeds 32s then it is timeout.

If Proxy server has insufficient resources to complete the processing of a request, it does not respond with the 100 Trying message. As a result of proxy overload, UAC does not receive the 100 Trying message. Since UAC does not receive the 100 Trying message, it retransmits INVITE message by timer T1. The retransmitted INVITE message leads to performance degradation.

Figure 2. Overload caused by retransmitted INVITE

SIP provides limited support for overload handling through the 503 response code, which tells an upstream element that it is overloaded. Overloaded proxy server sends the 503 message toward upstream server. If UAC receives the 503 response message, it must stay a state starting Timer D which is 32sec as default. Regulating making new calls by timer D, thus the load of network will fall down temporally.

Figure 3. Overload handling through 503 response code

3. Proposed Algorithm

A SIP Proxy Server acts as a router for SIP requests in SIP signaling network. In order to call session, All SIP message must be exchanged successfully. But the proxy server becomes overloaded with network congestion.

In this paper, we propose a new overload control algorithm based on priority scheduling to improve a call session setup delay and the throughput.

Figure 4 shows the queue structure of SIP proxy server. When proxy server receives SIP messages, SIP messages are classified as INVITE, Non-INVITE and the number of retransmitted INVITE, and then incoming SIP request messages are rejected with the 503 response message according to priority scheduling.

Figure 4. Queue structure of SIP Proxy server

The processing time of INVITE message is larger than that of Non-INVITE message. Therefore, we raised the priority of the Non-INVITE message in overload state. And the priority of INVITE message is shown in Figure 5.

As shown above, we considered the network congestion, call setup delay time and processing time to decide whether to drop the message or not.

Figure 6 is the example of the INVITE message, and the number of retransmitted INVITE message refers to CSeq of UAC.

More times for controlling SIP messages will be needed if network congestion becomes more frequent. If Proxy server has insufficient resources to complete the processing of a request, there is plenty of retransmitted SIP messages.

Call setup delay time is proportionate to the increased number of retransmitted INVITE message. Delay time of retransmitted INVITE message can be given as follows.

\[ I_i = T1 \times (2^i - 1) \]  
\[ b \] is the blocking probability of buffer and \( p_i \) is the success probability at I-th retransmission.

\[ p_i = (1 - b) \times b^{i-1} \]

The call setup delay from the first transmission to the success transmission D can be given as follows.
\[ D = \sum_{i=1}^{7} I_i \times P_i \]  

(3)

In this paper, we raised the priority of low re-transmitted INVITE message to improve a call setup delay.

\[ \text{INVITE Priority} \propto \frac{I}{T \times (2^I - 1)} \]  

(4)

This algorithm will shorten the call setup delay time and improve the throughput.

4. Performance Evaluation

We evaluated the performance of the proposed method compared with the previous overload control algorithm.

Previous algorithm 1: An overload control using two states of queue [7].

Previous algorithm 2: An overload control in proportion to the queue length of proxy server [9].

Proposed algorithm: An overload control based on priority scheduling.

Figure 7 shows the network topology which is bottleneck for performance evaluation.

We assume that the average SIP message arrival rate and service rate of the SIP proxy server. The offered load of the SIP proxy server \( \rho \) can be expressed as:

\[ \rho = \frac{\lambda}{\mu_{\text{SIP}}} \]  

(5)

Because the service rate of the SIP proxy server is constant, it has a relationship with the number of UA. Therefore we can adjust the number of UA.

The call setup delay from the first transmission to the success transmission \( D \) can be given as follows.

\[ \sum_{r=0}^{7} \frac{(1 - P_{\text{rej}}) \times I_r \times T \times (2^I - 1)}{\text{Total}} \]  

(6)

Figure 8 shows Call setup delay for dynamic load of the proxy server.
As shown in Figure 8, call setup delay decreases when the offered load exceeds 1.0 because of overload of the proxy server. As offered load increases, call setup delay increases because retransmitted INVITE messages are randomly processed.

The call setup delay of the proposed algorithm maintains the level. This is because of the Priority Scheduling flow of the INVITE message.

Figure 9 shows Throughput for dynamic load of proxy server. We compare the throughput performance for the different overload control algorithm. This is because we raise the priority of the Non-INVITE message. We can see that the throughput is improved significantly.

5. Conclusions

In this paper, we have proposed an Overload Control Algorithm based on Priority Scheduling for SIP Proxy Server. To prevent the increasing call setup delay, we propose Priority Scheduling flow of INVITE message to improve call setup delay in network congestion. Also, we raise the priority of the Non-INVITE message to improve throughput of proxy server. Evaluation results have shown that this algorithm is more efficient than prior work in Call setup delay, throughput.

6. References


